An Example of Failure Tolerant Operation of a Kinematically Redundant Manipulator

Christopher L. Lewis   Anthony A. Maciejewski

School of Electrical Engineering

Purdue University

West Lafayette, Indiana 47907

Abstract — The high cost involved in the retrieval and repair of robotic manipulators used for remediating nuclear waste, processing hazardous chemicals, or for exploring space or the deep sea, places a premium on the reliability of the system as a whole. For such applications, kinematically redundant manipulators are inherently more reliable since the additional degrees of freedom (DOF) may compensate for a failed joint. In this work, a redundant manipulator is considered to be fault tolerant with respect to a given task if it is guaranteed to be capable of performing the task after any one of its joints has failed and is locked in place. A method is developed for insuring the failure tolerance of kinematically redundant manipulators with respect to a given critical task. Techniques are developed for analyzing the manipulator's workspace to find regions which are inherently suitable for critical tasks due to their relatively high level of failure tolerance. Then, constraints are imposed on the range of motion of the manipulator to guarantee that a given task is completable regardless of which joint fails. These concepts are illustrated for a PUMA 560 that is used for a three-dimensional positioning task.

I. INTRODUCTION

Kinematically redundant manipulators have been proposed for use in the cleanup and remediation of nuclear and hazardous materials, as well as for remote applications such as deep space or sea exploration, where repair of broken actuators and sensors is impossible and the probability of their failure is increased due to the harsh operating environment [1, 2]. In these situations the extra degrees of freedom of a redundant manipulator may be used to compensate for the failed joints if the manipulator has been properly designed and controlled. The most basic task of a manipulator, i.e., the positioning/orienting of the end-effector in the workspace, is described by the forward kinematic equation

$$x = f(\theta), \qquad (1)$$

where $x \in R^m$ is the generalized vector of the position/orientation of the end-effector and $\theta \in R^n$ is the vector of joint variables. In this framework, point-to-point tasks can be described by a series of end-effector positions to be obtained at desired times, i.e., $x(t_i)$, with a kinematic inverse equation

$$\theta = f^{-1}(x) \qquad (2)$$

being solved to determine the corresponding required joint values, $\theta(t_i)$. A kinematically redundant manipulator can, in general, satisfy an end-effector positioning constraint, $x(t_i)$, with an infinite family of joint values satisfying (2). The underlying premise for advocating the use of redundant manipulators for critical applications is that if a joint should fail, then the redundancy of the manipulator may permit the completion of the task. Although commercial manipulators currently are not equipped with the necessary circuitry to detect failures and apply the brakes to any failing joint, the need for such a mechanism is well known [3]. If failed joints are locked then a single joint failure reduces the number of degrees of freedom (DOF) of the system by one and the new kinematic functions f and f^{-1} differ markedly from the original ones.

In [4] a method is described for designing manipulators to be fault tolerant with regard to a given point-to-point task. The authors assume that any joint may fail anywhere within its entire range of motion. A manipulator is said to be fault tolerant with respect to a given set of task points $x(t_i)$ only if there exist solutions to (2) for every possible failure in all joint configurations. With this assumption, the worst case typically occurs when a failing joint is folded in on itself. In the work described here, failure tolerance is achieved by imposing constraints on the motion of all joints prior to a failure.

By judiciously selecting the specific solution from the family of solutions to (2), the worst case need not occur. Thus failure tolerance may be achieved with less complex manipulator designs and for manipulators not originally designed with failure tolerance in mind.

An alternative to defining the manipulator's task as a sequence of end-effector positions is to specify the end-effector velocity profile. At the velocity level, the kinematic equations relating the joint rates $\dot{\theta}$ to the end-effector's velocity $\dot{x}$ are given by

$$\dot{x} = J\dot{\theta} \qquad (3)$$

where $J \in R^{m \times n}$ is the manipulator Jacobian matrix which is a function of the manipulator's configuration. The solution for all joint rates that satisfy the desired end-effector velocity can be represented by

$$\dot{\theta} = J^{+}\dot{x} + (I - J^{+}J)z \qquad (4)$$

where $^+$ indicates the pseudoinverse, $(I - J^{+}J)$ is the projection onto the null space, and z represents an arbitrary vector in the joint velocity space [5]. The second term in this equation clearly indicates that there is a family of joint trajectories that satisfy (3). However, unlike the kinematic function f relating the joint values to the end-effector's position, the Jacobian for the failed system is easily derived from the original system's Jacobian by zeroing the column of the failed joint. Using this fact it is possible to develop an inverse kinematic function which insures that the manipulator will have some degree of local dexterity after an arbitrary joint failure [6]. The measure of dexterity in this case is defined as the smallest singular value of the Jacobian, σ_m, so that a kinematic failure tolerance measure, kfm, is given by

$$kfm(\theta) = \min_{i=1 \text{ to } n} \sigma_m(^{i}J) \qquad (5)$$

where ^{i}J is the manipulator Jacobian matrix for the system with its i'th joint locked. Having a large value for kfm insures that after an arbitrary joint failure the manipulator will still be able to satisfy an arbitrary desired end-effector velocity in the vicinity of the failure. Unfortunately, this measure is inherently local in nature and can not guarantee that the complete trajectory remains feasible after the failure. However, it will be shown that this local failure tolerance measure can be used to guide the search for regions within the workspace for which one can insure that the entire desired task can be completed regardless of joint failures.

The remainder of this paper is organized as follows. First, a method for analyzing the fault tolerance of a given location in the workspace is discussed. Second, the constraints necessary to guarantee fault tolerance for a single point are described. Third, a procedure
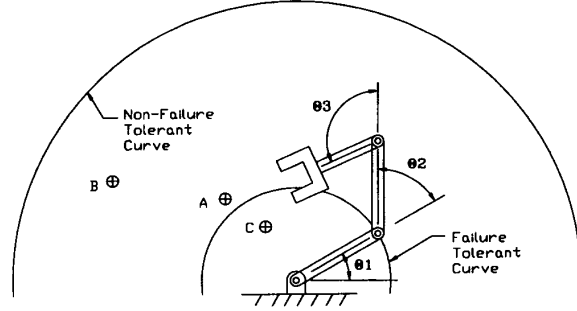

Fig. 1. A three degree-of-freedom planar manipulator with equal link lengths is shown with the curves in the workspace having maximum and minimum failure tolerance capabilities. The points A, B, and C are representative task space points that are analyzed for their global failure tolerance properties.

that uses the local measure of fault tolerance to identify candidate regions of the workspace where critical task should be placed is discussed. Then, a method for determining the constraints necessary to guarantee the fault tolerance of the manipulator with respect to the given critical path is outlined. Finally, the principles of this paper are demonstrated using a PUMA 560 manipulator to perform a three-dimensional positioning task.

II. SURFACES OF SELF-MOTION

For a kinematically redundant manipulator the family of joint configurations satisfying (1) forms an $(n-m)$-dimensional hyper-surface in the n-dimensional configuration space of the manipulator [7]. Joint motion constrained to this hyper-surface does not affect the position/orientation of the end-effector so that these hyper-surfaces are frequently referred to as self-motion manifolds. The null space of the manipulator's Jacobian given by the set of vectors satisfying (3) with $\dot{x} = 0$ defines the tangent hyperplane to the self-motion manifold. As a simple example, consider the 3 DOF planar manipulator shown in Fig. 1 for which the self-motion manifolds are one-dimensional curves. For this manipulator a projection of the self-motion curves onto the (θ_2, θ_3) plane is shown in Fig. 2. Each curve represents the family of joint variable combinations which place the end-effector at a constant radius from the base. From the figure, it is clear that some regions of the workspace have larger self-motion manifolds than others. For instance, consider the points corresponding to the reach singularity which occur when the arm is fully extended and the end-effector is at the boundary of its workspace. Each of these points is reachable by a single joint configuration which corresponds to the self-motion manifold also being a point. Clearly a workspace boundary
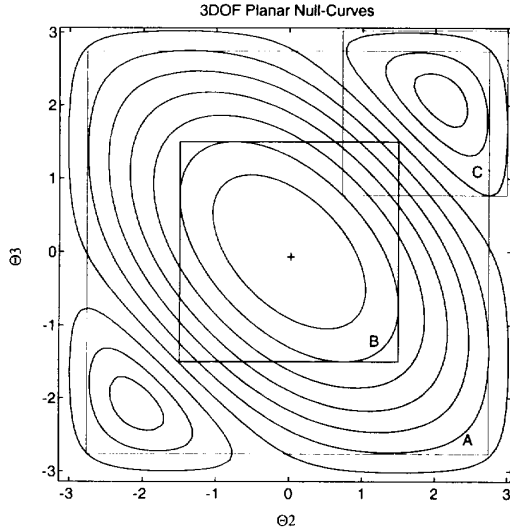
Fig. 2. The set of joint configurations that keep the manipulator's end-effector at a single location form curves in the configuration space of the manipulator. The curves shown are the self-motion curves for the 3 link planar manipulator depicted in Fig. 1. The self-motion curves for some regions of the workspace are markedly larger than others. Points with large self-motion curves tend to be more failure tolerant.

point will not be reachable after any joint failure unless the failure occurs with the end-effector at that point. In contrast, the workspace points exactly 1 meter from the base have self-motion manifolds which span the entire range of joint values for all three joints. In this case the failed manipulator will always be able to reach the entire set of points 1 meter from the base regardless of which joint fails, or the configuration in which it fails. It is interesting to note that the local failure tolerance measure (5) reaches its exact theoretically optimal value on the self-motion surface of this globally fault tolerant point. Also note that $kfm = 0$ at the boundary of the workspace since the Jacobian itself is singular at the reach singularity. These attributes lead to the use of kfm as a useful indicator for evaluating the workspace in order to place critical tasks. Clearly, when a joint fails and is locked, the manipulator is more likely to be able to reach points with large self-motion surfaces than those with small ones.

To guarantee that a manipulator is able to return to a desired workspace location one must, in general, constrain the motion range for each of the n joints. The minimum and maximum joint values of the ith joint, denoted $\theta_{i_{min}}$ and $\theta_{i_{max}}$, respectively, can be determined from the minimum and maximum values of θ_i over the entire self-motion manifold. This effectively superscribes an n-dimensional box aligned with the joint

axes around the self-motion manifold. The size of this bounding box is an indication of the inherent failure tolerance of the workspace point for which it was computed. If the manipulator fails while operating within the bounding box of a given desired end-effector location x^*, then it will always be able to position its end-effector at x^* regardless of where the end-effector is located when the failure occurs. For example, consider again the 3 DOF manipulator for which the bounding boxes associated with the self-motion surfaces for the three workspace points labeled A, B, and C in Fig. 1 have been drawn in Fig. 2. Note that although θ_1 and its associated boundaries are not shown, they also need to be considered. If the manipulator fails while within the boundary of any one of the bounding boxes, then the manipulator will always be able to position its end-effector at the associated point regardless of which joint fails. The region of the configuration space which lies inside all three bounding boxes is of particular interest. If the manipulator operates within this region, then regardless of which joint fails, it will be able to reach all three points. Unfortunately, if the joint motion is restricted to be within this region of configuration space then it is not possible to reach point C. However, after a failure, when the remaining joints are not restricted, the manipulator will be able to reach all three points. It should be clear that obtaining the bounding region for self-motion manifolds reveals the potential failure tolerance capabilities of various locations within the workspace.

It has been shown that the global fault tolerance associated with a point in the workspace is characterized by the self-motion manifold of the manipulator when its end-effector is at that point. Several iterative methods exist in the literature for characterizing one dimensional self-motion curves [8, 9, 10, 11]. For systems with two or more degrees of redundancy, an estimate of the size of the self-motion surface may be obtained using a Jacobian iteration of the form

$$\dot{\theta} = \pm(I - J^+ J)\hat{e}_i + J^+(x^* - x) \qquad (6)$$

where $\hat{e}_i$ is a unit vector along the ith joint axis and the error term $x^* - x$ is the difference between the desired end-effector position and its actual position. In practice, the first term provides motion along the self-motion manifold until the tangent hyperplane of the self-motion manifold becomes orthogonal to the joint axis direction $\hat{e}_i$, while the second term eliminates errors that could accumulated during the iterative procedure [10]. This method is effective for one-dimensional self-motion curves as in the 3DOF planar case but due to local extrema it may yield an insufficiently low estimate for self-motion manifolds of higher dimensions.
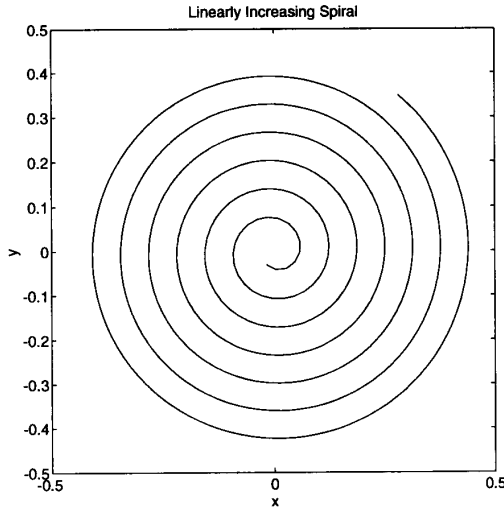
Fig. 3. A linearly increasing spiral passes within a controlled distance from every point in the plane and thus it may be used to estimate the bounds of a 2D surface in an n-dimensional space.
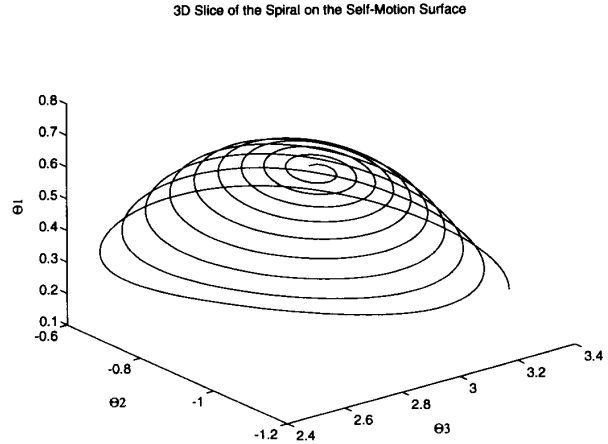

Fig. 4. For a three-dimensional positioning task the PUMA has two redundant degrees of freedom. Therefore it has the freedom to move its joints while holding its end-effector stationary. The spiral gives an indication of the two-dimensional surface embedded in the five-dimensional configuration space that describes how the first five joints of a PUMA can be moved without changing the three-dimensional position of the end-effector.

For a two-dimensional self-motion surface, a simple and effective method for estimating the bounds of the self-motion surface is to iteratively trace out a linearly increasing spiral on the self-motion surface. Keeping track of the values obtained by each joint along the spiral provides an estimate of the bounding box containing the self-motion surface. A 2D non-escaping spiral parameterized by the polar coordinates ϕ and r, depicted in Fig. 3, has the form

$$
\begin{aligned}
\dot{\phi} &= \frac{v}{r} \\
r &= \gamma\phi
\end{aligned}
\tag{7}
$$

where v is the speed along the spiral and γ controls the distance between successive rotations. Since this particular spiral passes within a controlled distance from every point in the plane, when it is transformed onto the self-motion surface it will tend to fill the surface. An iterative transformation procedure from parameter to configuration space is given by

$$
\dot{\theta} = \sin(\phi)\hat{v}_{n-1} + \cos(\phi)\hat{v}_n + J^+(x^* - x) \tag{8}
$$

where $\hat{v}_{n-1}$ and $\hat{v}_n$ are orthogonal unit vectors that span the null space of the manipulator's Jacobian evaluated at the current configuration. The vectors $\hat{v}_n$ and $\hat{v}_{n-1}$ can be computed as the singular vectors from the singular value decomposition (SVD) of J. But $\hat{v}_{n-1}$ and $\hat{v}_n$ are not unique since any two orthogonal vectors lying in the null space of the Jacobian are valid. As the joint configuration changes and the plane describing the null space rotates, the updated SVD arbitrarily selects

any two orthogonal vectors from the new rotated plane. To correctly transform the spiral, the null vectors are selected to be the unique vectors which are nearest in orientation to those in the previous iteration. For example, if the current singular vectors are represented by $\hat{v}_{n-1}$ and $\hat{v}_n$ then once (8) is evaluated and used to update the manipulator configuration, the new Jacobian will in general have different singular vectors $\hat{v}'_{n-1}$ and $\hat{v}'_n$. To accurately reflect the continuous rotation of these two vectors as the null space rotates, one can use the following set of equations

$$
\begin{aligned}
\hat{v}'_{n-1} &= \lambda\hat{w}_1 + (1-\lambda)\hat{w}_2 \\
\hat{v}'_n &= (1-\lambda)\hat{w}_1 - \lambda\hat{w}_2
\end{aligned}
\tag{9}
$$

where

$$
\lambda = \frac{(\hat{w}_1^T \hat{v}_{n-1})^2}{(\hat{w}_1^T \hat{v}_{n-1})^2 + (\hat{w}_2^T \hat{v}_{n-1})^2} \tag{10}
$$

and $\hat{w}_1$ and $\hat{w}_2$ are any unit vectors that span the new null space. Note that the sign should be examined to select the smallest resulting rotation. An ideal algorithm for computing the SVD that automatically calculates the continuous rotation of the null space is presented in [12]. An illustration of this technique for mapping out a two-dimensional self-motion surface is presented in Fig. 4. This figure shows a three-dimensional projection of the five-dimensional configuration space for a PUMA used in three-dimensional positioning tasks.

III. JOINT CONSTRAINTS TO GUARANTEE FAULT TOLERANCE

As was indicated in the previous section, a workspace location, x^*, may be guaranteed to be reachable regardless of joint failures if the manipulator is constrained to operate within the associated self-motion manifold's bounding box. This is evident since regardless of which joint fails, by definition, there must exist at least one alternative configuration on the self-motion manifold associated with x^* that corresponds to the joint value at which the joint failed. Therefore, the problem of maintaining the fault tolerance of a given critical location reduces to that of maintaining joint limits specified by the bounding box of the self-motion manifold for that location. This problem was first solved in [5] by using (4) and selecting z to result in motion away from the joint limits. The vector z may be computed by combining smooth functions so that the joint limits only affect the manipulator's motion when it is near the constraint boundaries [13].

For fault tolerance it is advantageous to locate critical task points in locations where the self-motion manifold bounds are large. For instance, jigs and fixtures should generally not be placed near the workspace boundaries since joint failures will render such regions unreachable. Although the tedious chore of measuring the size of the self-motion manifolds throughout the workspace could be done off-line, it has been found that the local measure of fault tolerance, $kfm(\theta)$, is a good indicator of size of the self-motion manifolds.

To insure that a task defined by a sequence of critical points may be performed regardless of joint failures, each point must be analyzed, the associated range of its self-motion surface determined, and then the intersection of the ranges for each point computed to determine the required joint constraints. Finally, it must be verified that the manipulator is able to reach each critical point while maintaining the constraints.

In summary, the following procedure is used to guarantee the failure tolerance of a redundant manipulator with respect to critical tasks. First, the workspace is analyzed using the local failure tolerance measure (5). Second, critical task are placed in regions of the workspace that have high values of local failure tolerance. Third, the bounding boxes for the self-motion surfaces associated with each critical location are determined using the procedures outlined in section II. Fourth, the intersection of the bounding boxes is calculated to determine the required constraints. Fifth, each critical workspace point is checked to determine if the manipulator is capable of positioning its end-effector at the desired location while maintaining the constraints imposed by the intersection of all bounding boxes. Fi-
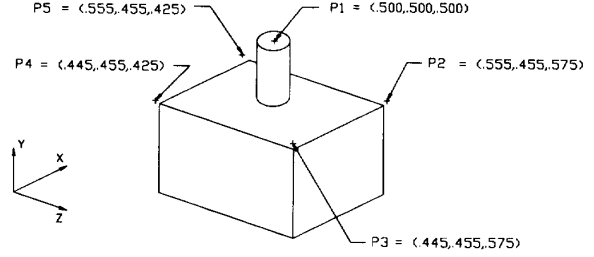

Fig. 5. A simple positioning path is used to demonstrate the failure tolerance of a PUMA 560 manipulator.

TABLE I
PUMA 560 DH Parameters

link	a	d	α	θ
1	0.0	0.0	−90	0
2	0.432	0.149	0	0
3	0.0	0.0	90	0
4	0.0	0.433	−90	0
5	0.0	0.0	90	0
6	0.0	0.392	0	0

nally, (4) is used with the joint limit constraints to insure the failure tolerance of the manipulator for the specified task.

IV. AN EXAMPLE USING A PUMA

To demonstrate the concepts outlined above, the following analysis was performed on a PUMA 560 manipulator at Sandia National Laboratory. The Denavit and Hartenberg parameters for the system are given in Table I. The task space is defined to be 3D positioning. Since the sixth joint of the PUMA only rotates the end-effector and does not change its position, the manipulator has nominally two redundant degrees of freedom with respect to the task space.

A simple positioning task defined by five points was chosen and is depicted in Fig. 5. To determine an ideal location in which to execute this task, the available workspace was analyzed using the local failure tolerance measure kfm defined by (5). The position of the first point $P1$ shown in Fig. 5 was selected for its relatively high local failure tolerance capabilities. A detailed description of local fault tolerance analysis is given in [6]. The self-motion surfaces for each of the five points were then examined using the spiral procedure outlined in section II. The resulting bounding box data is displayed in Table II. To illustrate the benefits of using the local failure tolerance measure for task placement,
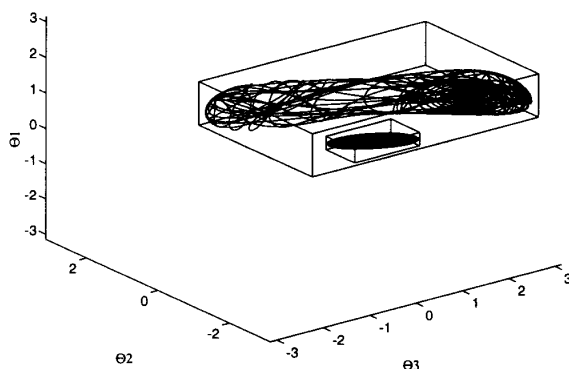
Fig. 6. Some locations in the PUMA's workspace allow for wider variations in the joints than others. With the end-effector's position fixed at a given location the range of motion for each joint may be determined using the projected spiral technique shown in Fig. 4. Here, two distinct workspace points are shown, one having a large self-motion surface and one with a small one, as indicated by their bounding boxes.

Fig. 6 displays the bounding box for the point $P1$ along with a representative location that has very poor failure tolerance capabilities. Completion of the task in a fault tolerant manner would not have been possible if it had been so poorly located. Next, with the motion of the joints constrained to lie within the intersection of the bounding boxes for all the points, it was verified that each point could be reached by iteratively solving (4) with the desired velocity being approximated by a position error until a solution was found (see Fig. 7). Now, with the constraints imposed, the manipulator is considered to be failure tolerant with respect to this path. To verify this (4) was implemented to trace out the trajectory through the points. Then, the technique was tested by simulating joint failures at random time intervals by locking a single joint. As expected, the manipulator was always able to complete the desired task.

To further demonstrate the advantages of performing the proposed analysis the point $P1$ was defined as a critical workcell location, e.g., a tool rest, which the manipulator must reach even after a joint failure. Using the bounding box of the self-motion surface for just this location, joint limits were imposed to guarantee the failure tolerance of the manipulator with respect to this location. First, the manipulator was moved to a point far away from the critical point without considering the effects of joint failure constraints. This configuration is shown in Fig. 8. The first joint was then locked and the manipulator attempted to move to $P1$. It could not

TABLE II
Self-Motion Surface Boundary Data

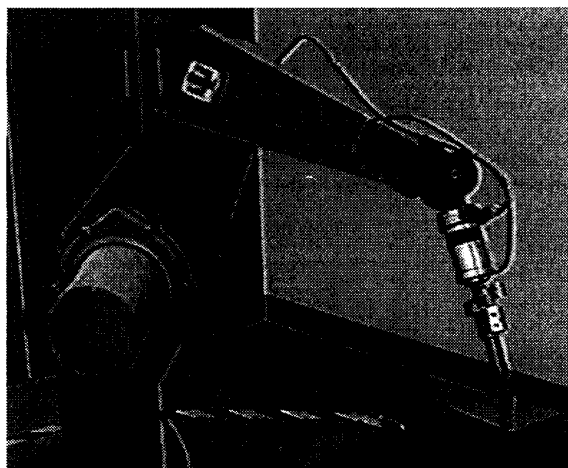|		p1	p2	p3	p4	p5	range
θ_1	min	-0.1	.0	-0.2	-.3	0.0	0.0
	max	1.1	1.1	1.0	1.2	1.3	1.0
θ_2	min	-1.4	-1.1	-1.2	-1.5	-1.3	-1.1
	max	1.8	1.6	1.8	2.2	1.9	1.6
θ_3	min	-0.9	-.6	-0.8	-1.1	-0.9	-.6
	max	4.0	3.7	3.9	4.2	4.0	3.7
θ_4	min	-6.3	-6.3	-6.3	-6.3	-6.3	-6.3
	max	6.3	6.3	6.3	6.3	6.3	6.3
θ_5	min	-2.5	-2.2	-2.4	-2.7	-2.5	-2.2
	max	2.5	2.2	2.4	2.7	2.5	2.2

Fig. 7. PUMA with its end-effector at the critical point in an optimal pose.

reach $P1$, but the configuration having the minimum position error is shown in Fig. 9. Next, with the failure tolerance constraints imposed, the manipulator is moved back to the same far away point. The configuration is quite different from before (see Fig. 10). Again the first joint is locked and the manipulator is asked to move its end-effector to $P1$. This time, as designed, it is able to reach $P1$ (see Fig. 11).

V. CONCLUSIONS

This paper has developed a method for insuring the failure tolerance of kinematically redundant manipulators. In this work, a redundant manipulator is considered to be fault tolerant with respect to a given task if it is guaranteed to be capable of performing the task
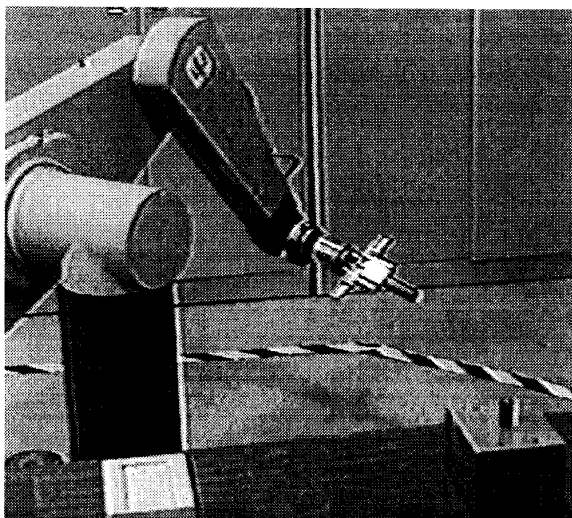
Fig. 8. PUMA with its end-effector at a point far away from the critical point in a configuration that was obtained without considering fault tolerance. Should a joint fail, it will not be able to position its end-effector at the critical point.
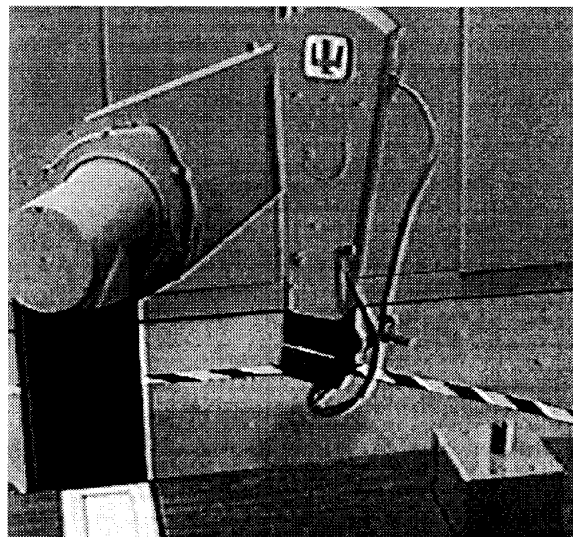

Fig. 10. PUMA with its end-effector at the same point as in Fig. 8 but in a configuration within the bounds of the self-motion surface bounding box. Regardless of which joint fails in this configuration, it will be able to position its end-effector at the critical point.
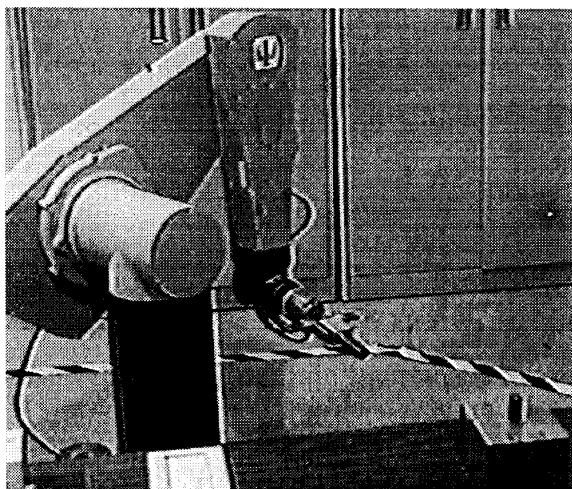

Fig. 9. PUMA with its end-effector moved from the non-fault tolerant configuration to its closest approach to the critical point after the first joint has failed.
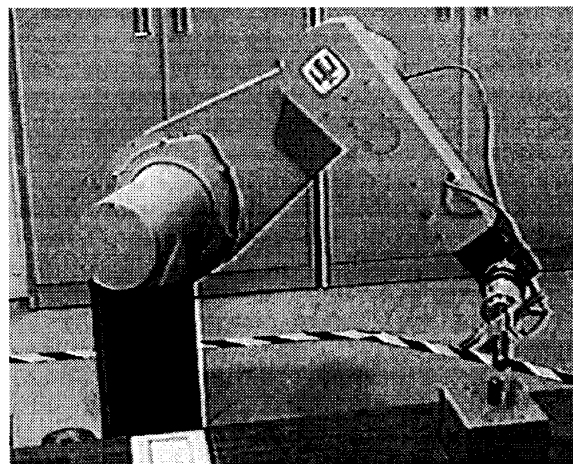

Fig. 11. PUMA with its end-effector moved from the failure tolerant configuration at the far point to the critical point while its first joint is locked. This demonstrates the failure tolerance of the configuration with respect to this critical point.

after any one of its joints has failed and is locked in place. Methods were developed for analyzing the manipulator's workspace to find regions which are inherently suitable for critical tasks due to their relatively high level of failure tolerance. Then, the required constraints were imposed on the range of motion of the manipulator to guarantee that a given task is completable regardless of arbitrary joint failures. These concepts were then illustrated for a PUMA 560 that was used for a three-dimensional positioning task.

REFERENCES

[1] B. Christensen, W. Drotning, and S. Thunborg, "Model-based, sensor-directed remediation of underground storage tanks," *Journal of Robotic Systems*, vol. 9, no. 2, pp. 145–159, 1992.

[2] R. Colbaugh and M. Jamshidi, "Robot manipulator control for hazardous waste-handling applications," *Journal of Robotic Systems*, vol. 9, no. 2, pp. 215–250, 1992.

[3] M. L. Visinsky, I. D. Walker, and J. R. Cavallaro, "Layered dynamic fault detection and tolerance for robots," *in Proc. 1993 International Conference on Robotics and Automation*, pp. 180–187, (Atlanta, Georgia), May 2-6, 1993.

[4] C. J. J. Paredis and P. K. Khosla, "Kinematic design of fault tolerant manipulators," *Computers and Electrical Engineering*, 1994.

[5] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-7, no. 12, pp. 868–871, December 1977.

[6] C. L. Lewis and A. A. Maciejewski, "Dexterity optimization of kinematically redundant manipulators in the presence of failures," *Computers and Electrical Engineering*, 1994.

[7] C. A. Klein and C. H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, no. 2, pp. 245–250, March/April 1983.

[8] J. W. Burdick, "On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifolds," *in Proc. 1989 International Conference on Robotics and Automation*, pp. 264–270, (Scottsdale, AZ), May 14-18, 1989.

[9] D. DeMers and K. Kreutz-Delgado, "Issues in learning global properties of the robot kinematic mapping," *in Proc. 1993 International Conference on Robotics and Automation*, pp. 205–212, (Atlanta, Georgia), May 2-6, 1993.

[10] C. A. Klein and B. E. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators," *International Journal of Robotics Research*, vol. 6, no. 2, pp. 72–83, Summer 1987.

[11] A. A. Maciejewski, "Kinetic limitations on the use of redundancy in robotic manipulators," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 205–210, April 1991.

[12] A. A. Maciejewski and C. A. Klein, "The singular value decomposition: Computation and applications to robotics," *International Journal of Robotics Research*, vol. 8, no. 6, pp. 63–79, December 1989.

[13] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *International Journal of Robotics Research*, vol. 4, no. 3, pp. 109–117, Fall 1985.